

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method of rendering a web page, the method comprising the computer-implemented steps of:
receiving user input via a user interface configured to receive a user-specified association between one or more page parameter names and one or more portlet parameter names;
based on the user input, generating and storing a mapping that maps the one or more page parameter names to the one or more portlet parameter names, wherein the mapping is stored separate from web pages associated with one or more page parameters that correspond to the one or more page parameter names;
wherein said web pages include the web page;
receiving a request to display the web page;
in response to receiving the request to display the web page, performing the steps of:
determining that the web page is associated with a particular page parameter that has a particular page parameter name from the one or more page parameter names;
using the mapping to determine which portlet parameter names are mapped to the particular page parameter name;
wherein using the mapping includes retrieving and inspecting the mapping to determine that the particular page parameter name is mapped to a particular portlet parameter name that corresponds to a particular portlet parameter of a portlet;

wherein the portlet is executable code that is operable to generate page components;
passing a value associated with the particular page parameter name to the portlet as a value of the particular portlet parameter that corresponds to the particular portlet parameter name;
the portlet generating a component based upon the value that is passed to the portlet as the value of the particular portlet parameter; and
inserting the component that was generated by the portlet into the web page;
wherein the steps of the method are performed by one or more computing devices.

2. (Previously Presented) The method of Claim 1, wherein:
using the mapping further includes determining that the particular page parameter name is mapped to a second portlet parameter name that corresponds to a second portlet parameter associated with a second component of the web page; and
in response to receiving the request to display the web page, further performing the step of passing the value associated with the particular page parameter name as the value of the second portlet parameter that is associated with a second portlet that generates the second component.
3. (Previously Presented) The method of Claim 1, wherein:
the step of generating and storing the mapping further comprises mapping a plurality of page parameter names, corresponding to page parameters for the web page, to a plurality of portlet parameter names corresponding to portlet parameters associated with the component of the web page;

the step of inspecting the mapping further comprises the step of determining which page parameter names of the plurality of page parameter names are mapped to each of the plurality of portlet parameter names;

the step of passing the value further comprises the step of passing, based on the mapping, values associated with the plurality of page parameter names as the values of the portlet parameters, corresponding to the plurality of portlet parameter names, of the portlet that generates the component; and

the step of the portlet generating the component further comprises the step of the portlet generating the component based upon the values associated with the plurality of portlet parameter names.

4. (Previously Presented) The method of Claim 1, wherein the step of generating and storing the mapping comprises the step of mapping the particular page parameter name to the particular portlet parameter name associated with the component of the web page without mapping the particular page parameter name to portlet parameter names associated with any other components of the web page.
5. (Previously Presented) The method of Claim 1, wherein the step of generating and storing the mapping comprises the steps of mapping the particular page parameter name to the particular portlet parameter name and mapping a second page parameter name to a second portlet parameter name corresponding to a second portlet parameter of the portlet that generates the component of the web page.

6. (Previously Presented) The method of Claim 1, further comprising the step of establishing for the particular page parameter name a default value, and wherein the step of passing the value associated with the particular page parameter name further comprises the step of passing the default value as the value of the particular portlet parameter of the portlet that generates the component.
7. (Previously Presented) The method of Claim 1, wherein the request to display the web page includes a URL and the URL includes the value associated with the particular page parameter name, and wherein the step of passing the value associated with the particular page parameter name is performed by passing the value contained in the URL as the value of the particular portlet parameter.
8. (Previously Presented) The method of Claim 1, further comprising the steps of:
presenting to a user a user interface for customizing the web page;
in response to the user interacting with the user interface, obtaining a user specified value
for the particular page parameter name; and
wherein the step of passing the value associated with the particular page parameter name
is performed by passing the user specified value as the value of the particular
portlet parameter of the portlet that generates the component.
9. (Previously Presented) The method of Claim 1, wherein a plurality of values are specified for the particular page parameter name and wherein:
the method further comprises the step of determining a selected value from the plurality
of values based on an override hierarchy; and

the step of passing further comprises the step of passing the selected value as the value of the particular portlet parameter of the portlet that generates the component.

10. (Previously Presented) The method of Claim 9, wherein the plurality of values includes a URL page parameter value and a customized page parameter value and the override hierarchy specifies that the URL page parameter value is the selected value.
11. (Previously Presented) The method of Claim 9, wherein the plurality of values includes a default page parameter value and a customized page parameter value and the override hierarchy specifies that the customized page parameter value is the selected value.
12. (Previously Presented) The method of Claim 9, wherein the plurality of values includes a default page parameter value and a portlet specified value and the override hierarchy specifies that the default page parameter value is the selected value.
13. (Previously Presented) The method of Claim 1, further comprising the step of presenting to a page designer a user interface for specifying the mapping between the particular page parameter name and the particular portlet parameter name.
14. (Previously Presented) The method of Claim 1, further comprising the step of registering the portlet with a portal repository, wherein the process of registering the portlet causes data associated with the portlet to be stored in the portal repository.

15. (Previously Presented) The method of Claim 14, wherein the data associated with the portlet is communicated to the portal repository as an XML document.
16. (Previously Presented) The method of Claim 1, further comprising the step of receiving input from a page designer, through a user interface, to create the mapping between the particular portlet parameter name and the particular page parameter name.
17. (Previously Presented) The method of Claim 1, wherein the value associated with the particular page parameter name is stored in memory and wherein:
the method further comprises the step of retrieving the stored value; and
the step of the portlet generating the component further comprises the step of the portlet generating the component based upon the retrieved value.
18. (Currently Amended) A method comprising the computer-implemented steps of:
receiving user input via a user interface configured to receive a user-specified association between one or more events and one or more actions and a user-specified association between one or more event output parameter names and one or more page parameter names;
based on the user input, generating and storing a first mapping that maps the one or more events to the one or more actions and the one or more event output parameter names to the one or more page parameter names, wherein the first mapping is stored separate from web pages associated with one or more page parameters that correspond to the one or more page parameter names;
wherein the web pages include a web page;

in response to a user manipulating a component of the web page, a portlet that previously generated the component generating a particular event;
wherein the portlet is executable code that is operable to generate page components;
logic associated with the web page intercepting data, passed by the portlet, that represents the particular event;
retrieving and inspecting the first mapping, wherein inspecting the first mapping includes:
determining, based on the first mapping and the intercepted data, an action to perform in response to the particular event;
based on the first mapping, determining that an event output parameter name, which corresponds to an event output parameter associated with the particular event, is mapped to a particular page parameter name; and
causing the action to be performed, wherein causing the action to be performed comprises passing a value associated with the event output parameter name as the value of a particular page parameter that corresponds to the particular page parameter name;
wherein the steps of the method are performed by one or more computing devices.

19. (Previously Presented) The method of Claim 18, wherein:
the web page is a first page and the particular page parameter is associated with a second page; and
the step of causing the action to be performed further comprises the step of passing the value of the particular page parameter to logic responsible for rendering the second page.

20. (Previously Presented) The method of Claim 18, wherein the step of causing the action to be performed further comprises the step of generating a request that specifies a URL, wherein the value of the particular page parameter is included in the URL.
21. (Original) The method of Claim 20, wherein:
the step of generating the request further comprises the step of generating a request for executable code; and
the step of causing the action to be performed further comprises the step of invoking the executable code.
22. (Original) The method of Claim 21, wherein the executable code is a web service.
23. (Previously Presented) The method of Claim 18, wherein:
the action comprises rendering a second page, wherein the particular page parameter is associated with the second page, and wherein rendering the second page comprises the steps of:
inspecting a second mapping to determine that the particular page parameter name is mapped to a particular portlet parameter name that corresponds to a particular portlet parameter of a second portlet that generates a second component of the second page that is based, at least in part, on the particular portlet parameter;
passing the value of the particular page parameter as the value of the particular portlet parameter, of the second portlet, that corresponds to the particular portlet parameter name;

the second portlet generating the second component based upon the value of the particular portlet parameter; and
inserting the second component that was generated by the second portlet into the second page.

24-46. (Canceled)

47. (Previously Presented) The method of Claim 1, wherein the portlet is a first portlet and wherein the mapping maps a single page parameter name, of the one or more page parameter names, to a first portlet parameter name corresponding to a first portlet parameter of the first portlet and to a second portlet parameter name corresponding to a second portlet parameter of a second portlet.

48. (Canceled)

49. (Currently Amended) A non-transitory computer-readable volatile or non-volatile medium storing one or more sequences of instructions for rendering a web page, which instructions when executed by one or more processors cause performance of steps comprising:
receiving user input via a user interface configured to receive a user-specified association between one or more page parameter names and one or more portlet parameter names;
based on the user input, generating and storing a mapping that maps the one or more page parameter names to the one or more portlet parameter names, wherein the

mapping is stored separate from web pages associated with one or more page parameters that correspond to the one or more page parameter names;

wherein said web pages include the web page;

receiving a request to display the web page;

in response to receiving the request to display the web page, performing the steps of:

- determining that the web page is associated with a particular page parameter that has a particular page parameter name from the one or more page parameter names;
- using the mapping to determine which portlet parameter names are mapped to the particular page parameter name;
- wherein using the mapping includes retrieving and inspecting the mapping to determine that the particular page parameter name is mapped to a particular portlet parameter name that corresponds to a particular portlet parameter of a portlet;
- wherein the portlet is executable code that is operable to generate page components;
- passing a value associated with the particular page parameter name to the portlet as a value of the particular portlet parameter that corresponds to the particular portlet parameter name;
- the portlet generating a component based upon the value that is passed to the portlet as the value of the particular portlet parameter; and
- inserting the component that was generated by the portlet into the web page;

wherein the steps of the method are performed by one or more computing devices.

50. (Previously Presented) The computer-readable medium of Claim 49, wherein:
the instructions that cause performance of the step of generating and storing the mapping
further comprise instructions which, when executed by the one or more
processors, cause performance of the step of determining that the particular page
parameter name is mapped to a second portlet parameter name that corresponds to
a second portlet parameter associated with a second component of the web page;
and
the instructions that cause performance of the steps in response to receiving the request to
display the page further comprise instructions which, when executed by the one or
more processors, cause the performance of the step of passing the value
associated with the particular page parameter name as the value of the second
portlet parameter that is associated with a second portlet that generates the second
component.
51. (Previously Presented) The computer-readable medium of Claim 49, wherein:
the instructions that cause performance of the step of generating and storing the mapping
further comprise instructions which, when executed by the one or more
processors, cause performance of the step of mapping a plurality of page
parameter names, corresponding to page parameters for the web page, to a
plurality of portlet parameter names corresponding to portlet parameters
associated with the component of the web page;
the instructions that cause performance of the step of inspecting the mapping further
comprise instructions which, when executed by the one or more processors, cause
performance of the step of inspecting the mapping to determine which page

parameter names of the plurality of page parameter names are mapped to each of the plurality of portlet parameter names;

the instructions that cause performance of the step of passing the value further comprise instructions which, when executed by the one or more processors, cause performance of the step of passing, based on the mapping, values associated with the plurality of page parameter names as the values of the portlet parameters, corresponding to the plurality of portlet parameter names, of the portlet that generates the component; and

the instructions that cause performance of the step of the portlet generating the component further comprise instructions which, when executed by the one or more processors, cause performance of the step of the portlet generating the component based upon the values associated with the plurality of portlet parameter names.

52. (Previously Presented) The computer-readable medium of Claim 49, wherein the instructions that cause performance of the step of generating and storing the mapping comprise instructions which, when executed by the one or more processors, cause performance of the step of mapping the particular page parameter name to the particular portlet parameter name associated with the component of the web page without mapping the particular page parameter name to portlet parameter names associated with any other components of the web page.

53. (Previously Presented) The computer-readable medium of Claim 49, wherein the instructions that cause performance of the step of generating and storing the mapping

comprise instructions which, when executed by the one or more processors, cause performance of the steps of mapping the particular page parameter name to the particular portlet parameter name and mapping a second page parameter name to a second portlet parameter name corresponding to a second portlet parameter of the portlet that generates the component of the web page.

54. (Previously Presented) The computer-readable medium of Claim 49, wherein the one or more sequences of instructions further comprise instructions which, when executed by the one or more processors, cause performance of the step of establishing for the particular page parameter name a default value, and wherein the instructions that cause performance of the step of passing the value associated with the particular page parameter name further comprise instructions which, when executed by the one or more processors, cause performance of the step of passing the default value as the value of the particular portlet parameter of the portlet that generates the component.
55. (Previously Presented) The computer-readable medium of Claim 49, wherein the request to display the web page includes a URL and the URL includes the value associated with the particular page parameter name, and wherein the instructions that cause performance of the step of passing the value associated with the particular page parameter name comprise instructions which, when executed by the one or more processors, cause performance of the step of passing the value contained in the URL as the value of the particular portlet parameter.

56. (Previously Presented) The computer-readable medium of Claim 49, wherein the one or more sequences of instructions further comprise instructions which, when executed by the one or more processors, cause performance of the steps of: presenting to a user a user interface for customizing the web page; and in response to the user interacting with the user interface, obtaining a user specified value for the particular page parameter name; wherein the instructions that cause performance of the step of passing the value associated with the particular page parameter name comprise instructions which, when executed by the one or more processors, cause performance of the step of passing the user specified value as the value of the particular portlet parameter of the portlet that generates the component.
57. (Previously Presented) The computer-readable medium of Claim 49, wherein a plurality of values are specified for the particular page parameter name and wherein: the one or more sequences of instructions further comprise instructions which, when executed by the one or more processors, cause performance of the step of determining a selected value from the plurality of values based on an override hierarchy; and the instructions that cause performance of the step of passing the value associated with the particular page parameter name further comprise instructions which, when executed by the one or more processors, cause performance of the step of passing the selected value as the value of the particular portlet parameter of the portlet that generates the component.

58. (Previously Presented) The computer-readable medium of Claim 57, wherein the plurality of values includes a URL page parameter value and a customized page parameter value and the override hierarchy specifies that the URL page parameter value is the selected value.
59. (Previously Presented) The computer-readable medium of Claim 57, wherein the plurality of values includes a default page parameter value and a customized page parameter value and the override hierarchy specifies that the customized page parameter value is the selected value.
60. (Previously Presented) The computer-readable medium of Claim 57, wherein the plurality of values includes a default page parameter value and a portlet specified value and the override hierarchy specifies that the default page parameter value is the selected value.
61. (Previously Presented) The computer-readable medium of Claim 49, wherein the one or more sequences of instructions further comprise instructions which, when executed by the one or more processors, cause performance of the step of presenting to a page designer a user interface for specifying the mapping between the particular page parameter name and the particular portlet parameter name.
62. (Previously Presented) The computer-readable medium of Claim 49, wherein the one or more sequences of instructions further comprise instructions which, when executed by the one or more processors, cause performance of the step of registering the

portlet with a portal repository, wherein the process of registering the portlet causes data associated with the portlet to be stored in the portal repository.

63. (Previously Presented) The computer-readable medium of Claim 62, wherein the data associated with the portlet is communicated to the portal repository as an XML document.
64. (Previously Presented) The computer-readable medium of Claim 49, wherein the one or more sequences of instructions further comprise instructions which, when executed by the one or more processors, cause performance of the step of receiving input from a page designer, through a user interface, to create the mapping between the particular portlet parameter name and the particular page parameter name.
65. (Previously Presented) The computer-readable medium of Claim 49, wherein the value associated with the particular page parameter name is stored in memory and wherein:
the one or more sequences of instructions further comprise instructions which, when executed by the one or more processors, cause performance of the step of retrieving the stored value; and
the instructions that cause performance of the step of the portlet generating the component further comprise instructions which, when executed by the one or more processors, cause the portlet to generate the component based upon the retrieved value.

66. (Currently Amended) A non-transitory computer-readable volatile or non-volatile medium storing one or more sequences of instructions which, when executed by one or more processors, cause performance of steps comprising:
- receiving user input via a user interface configured to receive a user-specified association between one or more events and one or more actions and a user-specified association between one or more event output parameter names and one or more page parameter names;
- based on the user input, generating and storing a first mapping that maps the one or more events to the one or more actions and the one or more event output parameter names to the one or more page parameter names, wherein the first mapping is stored separate from web pages associated with one or more page parameters that correspond to the one or more page parameter names;
- wherein the web pages include a web page;
- in response to a user manipulating a component of the web page, a portlet that previously generated the component generating a particular event;
- wherein the portlet is executable code that is operable to generate page components;
- ~~executing~~ logic associated with the web page to ~~intercept~~ intercepting data, passed by the portlet, that represents the particular event;
- retrieving and inspecting the first mapping, wherein inspecting the first mapping includes:
- determining, based on the first mapping and the intercepted data, an action to perform in response to the particular event;

based on the first mapping, determining that an event output parameter name,
which corresponds to an event output parameter associated with the
particular event, is mapped to a particular page parameter name; and
causing the action to be performed, wherein causing the action to be performed comprises
passing a value associated with the event output parameter name as the value of a
particular page parameter that ~~corresponds~~corresponds to the particular page
parameter name.

67. (Previously Presented) The computer-readable medium of Claim 66, wherein:
the web page is a first page and the particular page parameter is associated with a second
page; and
the instructions that cause performance of the step of causing the action to be performed
further comprise instructions which, when executed by the one or more
processors, cause performance of the step of passing the value of the particular
page parameter to logic responsible for rendering the second page.
68. (Previously Presented) The computer-readable medium of Claim 66, wherein the
instructions that cause performance of the step of causing the action to be performed
further comprise instructions which, when executed by the one or more processors, cause
performance of the step of generating a request that specifies a URL, wherein the value of
the particular page parameter is included in the URL.
69. (Previously Presented) The computer-readable medium of Claim 68, wherein:

the instructions that cause performance of the step of generating the request further
comprise instructions which, when executed by the one or more processors, cause
performance of the step of generating a request for executable code; and
the instructions that cause performance of the step of causing the action to be performed
further comprise instructions which, when executed by the one or more
processors, cause performance of the step of invoking the executable code.

70. (Previously Presented) The computer-readable medium of Claim 69, wherein the
executable code is a web service.
71. (Previously Presented) The computer-readable medium of Claim 66, wherein:
the action comprises rendering a second page, wherein the particular page parameter is
associated with the second page, and wherein rendering the second page
comprises:
inspecting a second mapping to determine that the particular page parameter name
is mapped to a particular portlet parameter name that corresponds to a
particular portlet parameter of a second portlet that generates a second
component of the second page that is based, at least in part, on the
particular portlet parameter;
passing the value of the particular page parameter as the value of the particular
portlet parameter, of the second portlet, that corresponds to the particular
portlet parameter name;
the second portlet generating the second component based upon the value of the
particular portlet parameter; and

inserting the second component that was generated by the second portlet into the
second page.

72. (Previously Presented) The computer-readable medium of Claim 49, wherein the portlet is a first portlet and wherein the mapping maps a single page parameter name, of the one or more page parameter names, to a first portlet parameter name corresponding to a first portlet parameter of the first portlet and to a second portlet parameter name corresponding to a second portlet parameter of a second portlet.